AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (currently amended) A data processing apparatus comprising:

processing ~~logic~~circuitry configured to perform a data processing operation on first and second data elements, the processing ~~logic~~circuitry comprising:

comparison ~~logic~~circuitry configured to compare at least a part of the first and second data elements in order to determine which of the first and second data elements is a larger data element, the comparison ~~logic~~circuitry being configured to produce a comparison result which has a first value if the first data element is the larger data element and a second value if the second data element is the larger data element;

absolute difference ~~logic~~circuitry configured to compute an absolute difference between a portion of the first data element and a portion of the second data element, the absolute difference ~~logic~~circuitry comprising:

adder ~~logic~~circuitry configured to invert one of said portions to produce an inverted data element portion and to add the inverted data element portion to the other of said portions and to the comparison result received from the comparison ~~logic~~circuitry in order to produce an intermediate result; and

output ~~logic~~circuitry configured to generate an inverted version of the intermediate result and to output as the absolute difference either the intermediate result or the inverted version of the intermediate result dependent on the comparison result,

wherein the adder circuitry is configured to invert the portion of the second data element, and wherein comparison circuitry is configured to set the comparison result as input to the adder

circuitry to a logical 0 value if the second data element is the larger data element and to a logical 1 value otherwise.

2. (currently amended) A data processing apparatus as claimed in Claim 1, wherein ~~the adder logic is configured to invert the portion of the second data element, and the comparison result is set to a logic 0 value if the second data element is the larger data element, and is set to a logic 1 value otherwise~~ the output ~~logic~~circuitry being configured to output as the absolute difference the inverted version of the intermediate result if the comparison result has a logic 0 value, and to output as the absolute difference the intermediate result if the comparison result has a logic 1 value.

3. (original) A data processing apparatus as claimed in Claim 1, wherein the comparison result is set to the first value if the first data element and the second data element have the same value.

4. (previously presented) A data processing apparatus as claimed in Claim 1, wherein the comparison logic is configured to perform a non-redundant subtract operation on said at least a part of the first and second data elements, and the comparison result comprises a carry out result of the non-redundant subtract operation.

5. (previously presented) A data processing apparatus as claimed in Claim 1, wherein the first and second data elements are floating point data elements, the first floating point data element specifying a first significand and the second floating point data element

specifying a second significand, and the absolute difference logic being configured to compute the absolute difference between the first significand and the second significand.

6. (original) A data processing apparatus as claimed in Claim 5, wherein each of the first and second floating point data elements comprise sign, exponent and fraction portions, the first and second significands being derived from the corresponding fraction portions of the first and second floating point data elements, the at least a part of the first and second data elements compared by the comparison logic comprising the exponent and fraction portions of the first and second floating point data elements.

7. (previously presented) A data processing apparatus as claimed in Claim 5, wherein the data processing apparatus is configured to receive first and second operands, the first operand comprising a plurality of said first floating point data elements, and the second operand comprising a corresponding plurality of said second floating point data elements, said comparison logic and said absolute difference logic being replicated within the data processing apparatus for each pair of first and second floating point data elements provided by the first and second operands.

8. (previously presented) A data processing apparatus as claimed in Claim 1, wherein the first and second data elements are integer data elements, and the portion of the first and second data elements that the absolute difference logic is configured to compute the absolute difference for is the entirety of the first and second integer data elements.

9. (original) A data processing apparatus as claimed in Claim 8, wherein the at least a part of the first and second data elements compared by the comparison logic comprises the entirety of the first and second integer data elements.

10. (previously presented) A data processing apparatus as claimed in Claim 8, wherein the data processing apparatus is configured to receive first and second operands, the first operand comprising a plurality of said first integer data elements, and the second operand comprising a corresponding plurality of said second integer data elements, said comparison logic being configured to receive the first and second operands and to produce, for each pair of first and second integer data elements provided by the first and second operands, an associated comparison result.

11. (previously presented) A data processing apparatus as claimed in Claim 10, wherein:

the absolute difference logic is configured to receive the first and second operands;

the adder logic is configured to invert one of the first and second operands to produce a plurality of inverted integer data elements and, for each pair of first and second integer data elements, to add the associated inverted data element to the other of the first and second data elements and to the associated comparison result received from the comparison logic in order to produce an associated intermediate result; and

the output logic is configured to generate an inverted version of each associated intermediate result and, for each pair of first and second integer data elements, to output as the

associated absolute difference either the associated intermediate result or the inverted version of the associated intermediate result dependent on the associated comparison result.

12. (currently amended) A method of operating a data processing apparatus to compute an absolute difference between a portion of a first data element and a portion of a second data element, comprising the steps of:

(a) comparing at least a part of the first and second data elements in order to determine which of the first and second data elements is a larger data element, and producing a comparison result which has a first value if the first data element is the larger data element and a second value if the second data element is the larger data element;

(b) employing adder ~~logic~~circuitry to invert one of said portions to produce an inverted data element portion and to add the inverted data element portion to the other of said portions and to the comparison result produced at said step (a) in order to produce an intermediate result; and

(c) generating an inverted version of the intermediate result and outputting as the absolute difference either the intermediate result or the inverted version of the intermediate result dependent on the comparison result,

wherein the adder circuitry inverts the portion of the second data element and step (a) comprises the step of setting the comparison result as input to the adder circuitry to a logic 0 value if the second data element is the larger data element, or setting the comparison result to a logic 1 value otherwise.

13. (currently amended) A method as claimed in Claim 12, wherein:

~~the adder logic inverts the portion of the second data element;~~

~~said step (a) comprises the step of setting the comparison result to a logic 0 value if the second data element is the larger data element, or setting the comparison result to a logic 1 value otherwise; and~~

said step (c) comprises the step of outputting as the absolute difference the inverted version of the intermediate result if the comparison result has a logic 0 value, and outputting as the absolute difference the intermediate result if the comparison result has a logic 1 value.

14. (original) A method as claimed in Claim 12, wherein at said step (a) the comparison result is set to the first value if the first data element and the second data element have the same value.

15. (original) A method as claimed in Claim 12, wherein said step (a) comprises the step of performing a non-redundant subtract operation on said at least a part of the first and second data elements, the comparison result comprising a carry out result of the non-redundant subtract operation.

16. (original) A method as claimed in Claim 12, wherein the first and second data elements are floating point data elements, the first floating point data element specifying a first significand and the second floating point data element specifying a second significand, and said steps (b) and (c) are performed to compute the absolute difference between the first significand and the second significand.

17. (original) A method as claimed in Claim 16, wherein each of the first and second floating point data elements comprise sign, exponent and fraction portions, the first and second significands being derived from the corresponding fraction portions of the first and second floating point data elements, the at least a part of the first and second data elements compared at said step (a) comprising the exponent and fraction portions of the first and second floating point data elements.

18. (previously presented) A method as claimed in Claim 16, wherein the data processing apparatus receives first and second operands, the first operand comprising a plurality of said first floating point data elements, and the second operand comprising a corresponding plurality of said second floating point data elements, said steps (a) to (c) being performed independently for each pair of first and second floating point data elements provided by the first and second operands.

19. (original) A method as claimed in Claim 12, wherein the first and second data elements are integer data elements, and the portion of the first and second data elements that the absolute difference is computed for is the entirety of the first and second integer data elements.

20. (original) A method as claimed in Claim 19, wherein the at least a part of the first and second data elements compared at said step (a) comprises the entirety of the first and second integer data elements.

21. (previously presented) A method as claimed in Claim 19, wherein the data processing apparatus receives first and second operands, the first operand comprising a plurality of said first integer data elements, and the second operand comprising a corresponding plurality of said second integer data elements, at said step (a) the first and second operands being received and, for each pair of first and second integer data elements provided by the first and second operands, the comparison performed at said step (a) producing an associated comparison result.

22. (original) A method as claimed in Claim 21, wherein:

at said step (b) the adder logic receives the first and second operands, the adder logic inverting one of the first and second operands to produce a plurality of inverted integer data elements and, for each pair of first and second integer data elements, adding the associated inverted data element to the other of the first and second data elements and to the associated comparison result produced at said step (a) in order to produce an associated intermediate result; and

said step (c) generates an inverted version of each associated intermediate result and, for each pair of first and second integer data elements, outputs as the associated absolute difference either the associated intermediate result or the inverted version of the associated intermediate result dependent on the associated comparison result.